AMENDMENTS TO THE CLAIMS

In the Claims:

Please amend claims 1, 3, 6, 8, 10, 13, 15, 17, 20, 22, 24, 27, 29, 31, 34, 36, 38, 41, 43, 44, 49, 50, 55, 56, 61, 62, 66, 67, 71, and 72, all as shown below. Applicant reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

Claims:

1. (Currently amended) A system for determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising:

an object temperature analyzer that accepts as input from a system developer a value for a limiting time,

wherein the object temperature analyzer determines for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time the object temperature analyzer sets the status of the object to cold, and if the length of time is less than the limiting time the object temperature analyzer sets the status of the object to warm; ~~the status of warm the objects and cold objects in said memory~~, and

wherein the object temperature analyzer determines [[the]] links [[between said]] from any of the warm objects [[and]] to any of the cold objects; and,

a report mechanism that reports information about [[said]] the links, for use by the system developer in determining potential memory leaks.

2. (Original) The system of claim 1 further comprising:

an object clusterer for clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

3. (Currently amended) The system of claim [[1]] 2 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

4. (Original) The system of claim 1 wherein the objects are used by the virtual machine.

5. (Original) The system of claim 1 wherein the links can be displayed on a computer screen device.

6. (Currently amended) The system of claim [[1]] 2 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

7. (Original) The system of claim 1 wherein the objects are not moved in memory when clustered.

8. (Currently amended) A system for determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising:

means for accepting as input from a system developer a value for a limiting time;

means for determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time, setting the status of the object to cold, and if the length of time is less than the limiting time, setting the status of the object to warm; ~~the status of warm objects and cold objects in said memory, and~~

means for determining [[the]] links [[between said]] from any of the warm objects to any of the [[and]] cold objects; and,

means for reporting information about [[said]] the links[[,]] for use by the system developer in determining potential memory leaks.

9. (Original) The system of claim 8 further comprising:

means for clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

10. (Currently amended) The system of claim [[8]] 9 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

11. (Original) The system of claim 8 wherein the objects are used by the virtual machine.

12. (Original) The system of claim 8 wherein the links can be displayed on a computer screen device.

13. (Currently amended) The system of claim [[8]] 9 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

14. (Original) The system of claim 8 wherein the objects are not moved in memory when clustered.

15. (Currently amended) A method for determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising the steps of:

accepting as input from a system developer a value for a limiting time;

determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time, setting the status of the object to cold, and if the length of time is less than the limiting time, setting the status of the object to warm; ~~the status of warm objects and cold objects in said memory, and~~

determining [[the]] links [[between said]] from any of the warm objects to any of the [[and]] cold objects; and,

reporting information about [[said]] the links[[,]] for use by the system developer in determining potential memory leaks.

16. (Original) The method of claim 15 further comprising the step of:

clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

17. (Currently amended) The method of claim [[15]] 16 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

18. (Original) The method of claim 15 wherein the objects are used by the virtual machine.

19. (Original) The method of claim 15 wherein the links can be displayed on a computer screen device.

20. (Currently amended) The method of claim [[15]] 16 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

21. (Original) The method of claim 15 wherein the objects are not moved in memory when clustered.

22. (Currently amended) A system for detecting memory leaks in an application server or run-time environment comprising:

a virtual machine executing within said run-time environment;

a memory space within said run-time environment for storing objects in memory, for use by a software application; and,

a temperature analyzer that accepts as input from a system developer a value for a limiting time,

wherein the temperature analyzer determines for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time, the object is marked as cold, and if the length of time is less than the limiting time, the object is marked as warm ~~the location of warm objects and cold objects in memory~~, and

wherein the temperature analyzer determines [[the]] links [[between said]] from any of the warm objects to any of the [[and]] cold objects, for use by the system developer in detecting memory leaks.

23. (Original) The system of claim 22 further comprising:

an object clusterer for clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

24. (Currently amended) The system of claim [[22]] 23 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

25. (Original) The system of claim 22 wherein the objects are used by the virtual machine.

26. (Original) The system of claim 22 wherein the links can be displayed on a computer screen device.

27. (Currently amended) The system of claim [[22]] 23 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

28. (Original) The system of claim 22 wherein the objects are not moved in memory when clustered.

29. (Currently amended) A system for detecting memory leaks in an application server or run-time environment comprising:

means for providing a virtual machine executing within said run-time environment;

means for storing objects in a memory, said objects for use by a software application; [[and,]]

means for accepting as input from a system developer a value for a limiting time;

means for determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time, marking the object as cold, and if the length of time is less than the limiting time, marking the object as warm; ~~the location of warm objects and cold objects stored in memory,~~ and

means for determining [[the]] links [[between said]] from any of the warm objects to any of the [[and]] cold objects, for use by the system developer in detecting memory leaks.

30. (Original) The system of claim 29 further comprising:

means for clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

31. (Currently amended) The system of claim [[29]] 30 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

32. (Original) The system of claim 29 wherein the objects are used by the virtual machine.

33. (Original) The system of claim 29 wherein the links can be displayed on a computer screen device.

34. (Currently amended) The system of claim [[29]] 30 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

35. (Original) The system of claim 29 wherein the objects are not moved in memory when clustered.

36. (Currently amended) A method for detecting memory leaks in an application server or run-time environment, comprising the steps of:

providing a virtual machine executing within said run-time environment;

storing objects in memory, for use by a software application; [[and,]]

accepting as input from a system developer a value for a limiting time;

determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time, marking the object as cold, and if the length of time is less than the limiting time, marking the object as warm; ~~the location of warm objects and cold objects stored in memory,~~ and

determining [[the]] links [[between said]] from any of the warm objects to any of the [[and]] cold objects, for use by the system developer in detecting memory leaks.

37. (Original) The method of claim 36 further comprising the step of:

clustering groups of warm objects to form warm clusters, and groups of cold objects to form cold clusters.

38. (Currently amended) The method of claim [[36]] 37 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

39. (Original) The method of claim 36 wherein the objects are used by the virtual machine.

40. (Original) The method of claim 36 wherein the links can be displayed on a computer screen device.

41. (Currently amended) The method of claim [[36]] 37 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

42. (Original) The method of claim 36 wherein the objects are not moved in memory when clustered.

43. (Currently amended) A system for providing potential memory leak information in a run-time environment, comprising:

an object temperature analyzer that accepts as input from a system developer a value for a limiting time, wherein the object temperature analyzer determines for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time the object temperature analyzer marks the object as cold, and if the length of time is less than the limiting time the object temperature analyzer marks the object as warm ~~the last access time of an object in memory~~;

an object clusterer that clusters groups of warm objects to form warm clusters and groups of cold objects to form cold clusters ~~together objects according to last access time~~; and

an object map that identifies links [[between]] from any of the warm objects in any of the warm clusters to any of the cold objects in any of the cold clusters ~~that have been~~

~~recently accessed, and other objects that have not been recently accessed,~~ to assist the system developer in determining potential memory leaks.

44. (Currently amended) The system of claim 43 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

45. (Original) The system of claim 43 wherein the objects are used by the virtual machine.

46. (Original) The system of claim 43 wherein the links can be displayed on a computer screen device.

47. (Original) The system of claim 43 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

48. (Original) The system of claim 43 wherein the objects are not moved in memory when clustered.

49. (Currently amended) A system for providing potential memory leak information in a run-time environment, comprising:

means for accepting as input from a system developer a value for a limiting time;

means for determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time the object temperature analyzer marks the object as cold, and if the length of time is less than the limiting time the object temperature analyzer marks the object as warm ~~the last access time of an object in memory~~;

means for clustering groups of warm objects to form warm clusters and groups of cold objects to form cold clusters ~~the objects according to last access time~~; and,

means for identifying links [[between]] from any of the warm objects in any of the warm clusters to any of the cold objects in any of the cold clusters ~~recently accessed objects and not recently accessed objects~~ to assist the system developer in determining potential memory leaks.

50. (Currently amended) The system of claim 49 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

51. (Original) The system of claim 49 wherein the objects are used by the virtual machine.

52. (Original) The system of claim 49 wherein the links can be displayed on a computer screen device.

53. (Original) The system of claim 49 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

54. (Original) The system of claim 49 wherein the objects are not moved in memory when clustered.

55. (Currently amended) A method for providing potential memory leak information in a run-time environment, comprising the steps of:

accepting as input from a system developer a value for a limiting time;

determining for each object whether the object has persisted in memory without being accessed or referenced for a length of time greater than the limiting time, wherein if the length of time is greater than the limiting time the object temperature analyzer marks the object as cold, and if the length of time is less than the limiting time the object temperature analyzer marks the object as warm ~~the last access time of an object in memory~~;

clustering groups of warm objects to form warm clusters and groups of cold objects to form cold clusters ~~the objects according to last access time~~; and,

identifying links [[between]] from any of the warm objects in any of the warm clusters to any of the cold objects in any of the cold clusters ~~recently accessed objects and not recently accessed objects~~ to assist the system developer in determining potential memory leaks.

56. (Currently amended) The method of claim 55 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

57. (Original) The method of claim 55 wherein the objects are used by the virtual machine.

58. (Original) The method of claim 55 wherein the links can be displayed on a computer screen device.

59. (Original) The method of claim 55 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

60. (Original) The method of claim 55 wherein the objects are not moved in memory when clustered.

61. (Currently amended) A system for use in determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising:

objects located in the memory of a run-time environment, wherein each object includes a time stamp field and a time stamp therein, and wherein the time stamp is updated with a current system time T_{access} when the object is accessed or referenced;

an object temperature analyzer that accepts as input from a system developer a value for a limiting time T_{limit}, wherein the object temperature analyzer after a time T_{check}, marks each object as warm if the length of time between T_{access} and T_{check} is less than the limiting time T_{limit} or marks the object as cold if the length of time between T_{access} and T_{check} is greater than the limiting time T_{limit} ~~being either warm or cold~~;

an object clusterer that clusters groups warm objects [[together as]] to form warm clusters and groups cold objects [[together as]] to form cold clusters; and,

a display device that displays an object map, the object map including links [[between]] from any of the warm objects in any of the warm clusters [[and]] to any of the cold objects in any of the cold clusters.

62. (Currently amended) The system of claim 61 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

63. (Original) The system of claim 61 wherein the objects are used by the virtual machine.

64. (Original) The system of claim 61 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

65. (Original) The system of claim 61 wherein the objects are not moved in memory when clustered.

66. (Currently amended) A system for use in determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising:

means for creating objects in the memory of a run-time environment;

means for stamping each object with a time stamp T_{init} when created;

means for updating the time stamp of each object with a current system time T_{access} when [[as each]] the object is accessed or referenced;

means for accepting as input by a system developer a value for a limiting time T_{limit};

means for after a time $T_{check,}$ marking each object as warm if the length of time between T_{access} and T_{check} is less than the limiting time T_{limit} or marking the object as cold if the length of time between T_{access} and T_{check} is greater than the limiting time T_{limit} ~~being either warm or cold~~;

means for clustering groups of warm objects [[together]] to form warm clusters and groups of cold objects [[together]] to form cold clusters; and,

means for displaying an object map showing links [[between]] from any of the warm objects in any of the warm clusters [[and]] to any of the cold objects in any of the cold clusters.

67. (Currently amended) The system of claim 66 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

68. (Original) The system of claim 66 wherein the objects are used by the virtual machine.

69. (Original) The system of claim 66 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

70. (Original) The system of claim 66 wherein the objects are not moved in memory when clustered.

71. (Currently amended) A method for use in determining potential memory leaks in a run-time environment, said run-time environment including a virtual machine and a memory space for storing objects, comprising:

creating objects in the memory of a run-time environment;

stamping each object with a time stamp T_{init} when created;

updating the time stamp of each object with a current system time T_{access} when [[as each]] the object is accessed or referenced;

accepting as input by a system developer a value for a limiting time T_{limit};

after a time T_{check}, marking each object as warm if the length of time between T_{access} and T_{check} is less than the limiting time T_{limit} or marking the object as cold if the length of time between T_{access} and T_{check} is greater than the limiting time T_{limit} ~~being either warm or cold~~;

clustering groups of warm objects [[together]] to form warm clusters and groups of cold objects [[together]] to form cold clusters; and,

displaying an object map showing links [[between]] from any of the warm objects in any of the warm clusters [[and]] to any of the cold objects in any of the cold clusters.

72. (Currently amended) The method of claim 71 wherein the links include[[s]] any or both of warm object - cold object links and warm cluster - cold cluster links.

73. (Original) The method of claim 71 wherein the objects are used by the virtual machine.

74. (Original) The method of claim 71 wherein the limiting time determining whether an object is warm or cold can be adjusted by the developer to better distinguish between warm and cold objects or warm and cold clusters.

75. (Original) The method of claim 71 wherein the objects are not moved in memory when clustered.